



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/293,737	04/16/1999	GORDON B. DOW	07844-315001	5291
21876 7590 09/07/2007 FISH & RICHARDSON P.C. P.O. Box 1022 MINNEAPOLIS, MN 55440-1022			EXAMINER ANYA, CHARLES E	
			ART UNIT 2194	PAPER NUMBER
			MAIL DATE 09/07/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

mn

Office Action Summary

Application No.

09/293,737

Applicant(s)

DOW, GORDON B.

Examiner

Charles E. Anya

Art Unit

2194

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3/MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 20 June 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-9, 11, 13-21, 28, 29, 31-33, 35, 36, 38-40, 42-55 and 58-77 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-9, 11, 13-21, 28, 29, 31-33, 35, 36, 38-40, 42-55 and 58-77 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.


WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 1-9,11,13-21,28,29,31-33,35,36,38-40,42-55,58 and 59-77 are pending in this application.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. **Claims 1-9,11,13-18,28,29,31,32,35,36,38,39,42-53 and 62-73 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,929,864 to Picott et al. in view of U.S. Pat. No. 5,404,428 to Wu.**

4. As to claim 1, Picott teaches a method in a computer program for maintaining dependencies among a set of objects each having a value, the set of objects including an object A and an object B, the method for maintaining dependencies comprising: when the value of object A is a function of the value of object B ("...dependent node A..." Col. 7 Ln. 31 – 36) and the value of object B changes, marking object A as dirty ("..."you are dirty"..." Col. 7 Ln. 35 – 39) and when the value of object B changes,

Art Unit: 2194

invalidating the dependents of object B (“...“you are dirty”...” Col. 7 Ln. 35 – 39), including severing dependencies among the dependent of object B and all of their further dependents (“...redraw...” Col. 7 Ln. 35 – 39: NOTE: although severing dependencies are not explicitly taught, it is inherently taught since the node of the directed acyclic graph is invalidated when the node changes and a **redraw** request is initiated. The redraw request indicates that the original directed acyclic graph (i.e. the directed acyclic graph before the change) has been discarded).

Picott is silent with reference to not recomputing the value of object A until object A is queried for a value when the value of object B changes, invalidating the all of their further dependents of object B and causing each invalidated observer-only object to recompute its value by querying the values of the objects from which the observer-only object depend.

Wu teaches not recomputing the value of object A until object A is queried for a value when the value of object B changes, invalidating the all of their further dependents of object B (“...until...” Col. 8 Ln. 24 – 26, “...invalidated...No calculation...” Col. 9 Ln. 1 – 10) and causing each invalidated observer-only object to recompute its value by querying the values of the objects from which the observer-only object depends (“...calculating invalid item(s)...” Col. 9 Ln. 10 – 61).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system Picott with the teaching of Wu because the teaching of Wu would improve the system of Picott by optimizing evaluation traversal of a directed acyclic graph (Col. 9 Ln. 42 - 47).

5. As to claim 2, Picott teaches the method of claim 1, further comprising: providing object B in the construction of object A, wherein the value of object A is a function of the value of the object B that was provided in the construction of object A (Col. 7 Ln. 31 – 36).

6. As to claim 3, Wu teaches the method of claim 1, further comprising: providing in object B a `handleRequest` method that adds a requester owned by object A to a dependents list for object B, the dependents list identifying all objects whose value is a function of the value of object B (“...dependency...” Col. 9 Ln. 56 – 61).

7. As to claim 4, Wu teaches the method of claim 3, wherein the dependents lists for all objects in the set collectively define a directed, acyclic dependency graph (“...acyclic directed graph...” Col. 9 Ln. 56 – 67).

8. As to claim 5, Wu teaches the method of claim 1, further comprising: when an object is marked as dirty, breaking any dependency relationships the marked object may have had; and when the value of an object is recomputed, identifying the objects on which the recomputed value is actually dependent and identifying the recomputed object as dependent only on the identified objects (Col. 9 Ln. 1 - 47).

9. As to claim 6, Wu teaches the method of claim 1, wherein the set of objects includes settable objects and dependent objects (“...directed acyclic graph....” Col. 3 Ln. 8 – 21), and each dependent object maintains a flag whose setting marks the dependent object as valid or invalid or dirty (“...flag...” Col. 3 Ln. 19 - 24).

10. As to claim 7, Picott teaches a method in a computer program for maintaining dependencies among a set of objects each having a value, the method for maintaining dependencies comprising (“...dependency graph...” Col. 1 Ln. 55 – 67); identifying the objects upon which a given object depends as those objects into which the given object passed itself as a requester during execution of a compute method of the given object (“...“evaluate me”...” Col. 7 Ln. 35 – 47), and marking the given object as dirty whenever the value of any one of the identified objects changes (“...“you are dirty”...” Col. 7 Ln. 35 – 39) and modifying a dependency graph by severing dependencies among dependents of the identified objects with changed value and all of their further dependents to create a modified dependency graph (“...redraw...” Col. 7 Ln. 35 – 39: NOTE: although severing dependencies are not explicitly taught, it is inherently taught since the node of the directed acyclic graph is invalidated when the node changes and a **redraw** request is initiated. The redraw request indicates that the original directed acyclic graph (i.e. the directed acyclic graph before the change) has been discarded).

Picott is silent with reference to not recomputing the value of the given object until the given object is queried for a value.

Wu teaches not recomputing the value of the given object until the given object is queried for a value ("...until..." Col. 8 Ln. 24 – 26, "...invalidated... No calculation..." Col. 9 Ln. 1 – 10).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system Picott with the teaching of Wu because the teaching of Wu would improve the system of Picott by optimizing evaluation traversal of a directed acyclic graph (Col. 9 Ln. 42 - 47).

11. As to claim 8, Picott teaches the method of claim 7, further comprising: identifying as dependents of a root object all objects that passed themselves as requester objects to the root object or to a dependent of the root object during execution of the requester objects respective compute methods, whereby the set of dependents of the root object is a set that changes based on the computation of dependents and not the root object itself ("... "evaluate me"..." Col. 7 Ln. 35 – 47).

12. As to claims 9, 46 and 48, see the rejection of claim 6 above.

13. As to claim 11, Wu teaches a method for changing objects having values defining state of a computer program application, comprising: receiving a change to a value of a changed object, the changed object having objects depending directly on the changed object and objects depending indirectly on the changed object through an object different from the changed object (Step 602 Col. 23 Ln. 21 – 31), the changed object

being a settable object in the compute program application (VDC window..." 8 Ln. 18 – 33); registering the change with a transaction ("...performs the requested changes..." Col. 23 Ln. 21 – 26); dirtying all objects dependent (directly or indirectly) on the changed object (Step 604 Col. 23 Ln. 21 – 26) and whenever a leaf object is encountered as a dependent object, enqueueing the leaf object for synchronization after the transaction is committed (Step 605 Col. 23 Ln. 28 – 31).

Picott teaches severing dependencies from the changed object and all of its direct and indirect dependent objects ("...redraw..." Col. 7 Ln. 35 – 39: NOTE: although severing dependencies are not explicitly taught, it is inherently taught since the node of the directed acyclic graph is invalidated when the node changes and a **redraw** request is initiated. The redraw request indicates that the original directed acyclic graph (i.e. the directed acyclic graph before the change) has been discarded.

14. As to claim 13, Wu teaches the method of claim 11, wherein leaf object synchronization comprises: recomputing a value for each objects marked as dirty, identifying the objects on which the recomputed value is actually dependent, and identifying the recomputed object as dependent only on the identified objects ("...calculations may be performed..." Col. 9 Ln. 8 – 41, "...calculated back..." Col. 10 Ln. 47 – 55, figure 7 Col. 23 Ln. 32 – 67).

15. As to claim 14, Wu teaches the method of claim 13, further comprising: using a requester object to make the transaction consistent ("...request..." Col. 9 Ln. 8 – 10),

Art Unit: 2194

the requester object operating to request an object's value so that the requested value cannot change until the requester terminates, at which time all objects whose values were requested by the requester object are released ("...evaluated lazily..." Col. 8 Ln. 22 – 26, Col. 8 Ln. 24 – 26, "...invalidated... No calculation..." Col. 9 Ln. 1 – 10).

16. As to claim 15, Wu teaches a method for changing objects defining stat of a computer program application, comprising: creating a transaction and registering the transaction, one or more changes to settable objects, each change being made to a corresponding changing object (Step 602 Col. 23 Ln. 21 – 31, "...performs the requested changes..." Col. 23 Ln. 21 – 26); for each change registered, traversing a dependency graph from the changing object and (i) for each dependent object on the dependency graph, marking the dependent object as dirty ("...invalidates..." Col. 8 Ln. 27 – 33, Step 604 Col. 23 Ln. 21 – 26) and, (ii) accumulating each leaf object encountered in traversing the dependency graph in a strobe queue ("...set of bits..." Col. 8 Ln. 30 – 33, Step 605 Col. 23 Ln. 28 – 31); and traversing the strobe queue after all changes to settable objects have been registered and synchronized each leaf object be recomputing values for objects marked as dirty ("...calculations may be performed..." Col. 9 Ln. 8 – 41, "...calculated back..." Col. 10 Ln. 47 – 55, figure 7 Col. 23 Ln. 32 – 67).

Picott teaches detaching the dependent object from the dependency graph and rejoining recomputed objects with the dependency graph, whereby leaf objects are rejoined with the dependency graph ("...redraw..." Col. 7 Ln. 35 – 39; NOTE: although

severing dependencies are not explicitly taught, it is inherently taught since the node of the directed acyclic graph is invalidated when the node changes and a **redraw** request is initiated. The redraw request indicates that the original directed acyclic graph (i.e. the directed acyclic graph before the change) has been discarded).

17. As to claim 16, Wu teaches the method of claim 15, wherein: the dependency graph represents application state ("Context state..." Col. 8 Ln. 11 – 17); the roots of the dependency graph are the settable objects of the application state (VDC window..." Col. 8 Ln. 11 – 26) and the intermediate nodes of the dependency graph are dependent objects whose values are the results of intermediate computations (Col. 15 Ln. 46 - 67, Col. 18 Ln. 63 - 67).

18. As to claim 17, Wu teaches the method of claim 15, wherein: the leaf objects of the dependency graph are coupled to a user interface (figure 4 Col. 7 Ln. 35 – 67).

19. As to claims 18, 72 and 73, see the rejection of claim 17 above.

20. As to claims 28 and 35, see the rejection of claim 1 above.

21. As to claims 29 and 36, see the rejection of claim 7 above.

22. As to claims 31 and 38, see the rejection of claim 11 above.

23. As to claims 32 and 39, see the rejection of claim 15 above.
24. As to claims 42 and 45, see the rejection of claims 2 and 5 respectively.
25. As to claims 43 and 44, see the rejection of claims 3 and 4 respectively.
26. As to claims 47 and 67, see the rejection of claim 8 above.
27. As to claims 49,50 and 51, see the rejection of claims 13,14 and 16 respectively.
28. As to claims 52 and 53, see the rejection of claim 17 above.
29. As to claims 62-66, see the rejection of claims 2-6 respectively.
30. As to claims 68-71, see the rejection of claims 9,13,14 and 16 respectively.
31. **Claims 19-21,33,40,54-57,74 and 75 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,404,428 to Wu in view of U.S. Pat. No. 5,526,475 to Razdow.**

32. As to claim 19, Wu teaches a method in computer program for managing dependency among a set of objects, each object of the set of objects having a value, the set of objects including dependent objects, wherein a given object having objects depending directly on the given object and objects depending indirectly on a given object through other objects of the set of objects (“...acyclic graph...” Col. 3 Ln 8 – 19, Ln. 63 – 67, Col. 8 Ln. 63 – 67, Col. 9 Ln. 1 – 10) and each dependent object having a value that is a function of the values of one or more of the other objects in the set of objects (“View Model Derived Data...” Col. 6 Ln. 44 – 67).

Wu is silent with reference to the method comprising: calculating the dependency among objects in the set of objects dynamically at the time objects calculate their values and a given object having objects depending directly on the given object.

Razdow teaches the method comprising: calculating the dependency among objects in the set of objects dynamically at the time objects calculates their values (Col. 4 Ln. 63 – 67, Col. 5 Ln. 1 – 10, “...automatically...” Col. 9 Ln. 8 – 26).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Wu with the teaching of Radow because the teaching of Radow would improve the system of Wu by providing a method for automatically updating an object tree (Radow Col. 5 Ln. 1 – 10).

33. As to claim 20, Wu teaches the method of claim 19, wherein each observed object in the set of objects has one or more accessor methods that each take a requester argument and return a current value of the observed object, the requester

Art Unit: 2194

argument identifying the object requesting the value of the observed object

("...functions... evaluation of the items..." Col. 7 Ln. 3 – 12, Ln. 42 – 46, "...returning..."

Col. 7 Ln. 58 – 66).

34. As to claim 21, Wu teaches the method of claim 19, wherein each settable object in the set of objects has a value setting method that takes two arguments, namely a transaction argument identifying a transaction with which the change to the settable object's value is registered and a new value for the settable object ("...function object_set()..." Col.86 Ln. 18 – 26).

35. As to claims 33 and 40, see the rejection of claim 19 above.

36. As to claims 54 and 74, see the rejection of claim 20 above.

37. Claims 58,59,76 and 77 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,404,428 to Wu in view of U.S. Pat. No. 5,526,475 to Razdow as applied to claim 19 above, and further in view of U.S. Pat. No. 5,815,415 to Bentley et al.

38. As to claim 58, Radow and Wu is silent with reference to the method of claim 19, wherein all objects of the set are instantiated from object-oriented programming classes that inherit a set of methods from a common base class.

Bentley teaches the method of claim 19, wherein all objects of the set are instantiated from object-oriented programming classes that inherit a set of methods from a common base class (Object 52 Col. 23 Ln. 17 – 25).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Radow and Wu with the teaching of Bentley because the teaching of Bentley would improve the system of Radow and Wu by providing an object-based computerized modeling systems that allows information to be shared with other object-based programs by publishing appropriate interfaces (Bentley Col. 5 Ln. 37 – 39).

39. As to claim 59, Bentley teaches the method of claim 58, wherein the common base class is a requestor class with method to lock down and reset queried values in order to guarantee consistency (“...maintain the integrity...” Col. 4 Ln. 6 – 14, Col. 40 Ln. 35 – 39).

40. As to claims 76 and 77, see the rejection of claims 58 and 59 respectively.

Response to Arguments

Applicant's arguments filed 6/20/07 have been fully considered but they are not persuasive.

Applicant argues in substance that (1) that 101 rejection should be withdrawn because claims 19-21,33 and 54-55 are directed to statutory subject matter, (2) the

Art Unit: 2194

Picott prior art does not teach “severing dependencies among the dependents of object B and all of their further dependents when the value of object B changes”, (3) the Picott prior art does not teach “identifying the objects upon which a given object depends as those objects into which the given object passed itself as a requester during execution of a compute method of the given object”, (4) no reference is cited that teaches the claim limitation (claim 7) “modifying a dependency graph by severing dependencies among dependents of the identified objects with changed value and all of their further dependents to create a modified dependency graph”, (5) the Picott prior art does not teach “detaching dependent objects from the dependency graph and (6) the Razdow prior art does not teach “calculating the dependency among objects in the set of objects dynamically at the time objects calculate their values”.

The Examiner respectfully traverses Applicant’s arguments:

As to point (1), after review of Applicant’s argument and the referenced claims the Examiner is withdrawing the 101 rejection.

As to point (2), it may be best to first explain the meaning of the word “redraw”. The web site “http://en.wikipedia.org/wiki/Main_Page” describes “redraw” as follows: “To draw **again**”. The implication of this description is that for something to be drawn **again** it must have been drawn (or existed) before and needs to be changed for some reason. The redraw request of the Picott prior art resulted from a change in the directed acyclic graph (DAG). When a change or modification to an **existing DAG** occurs the redraw request is initialed in order to update the existing DAG. Although “severing dependencies among the dependents of object B and all of their further dependents

Art Unit: 2194

when the value of object B changes” is not explicitly taught, it is inherently taught because the directed acyclic graph is invalidated when the changes occur and a **redraw** request is initiated. The redraw request indicates that the original directed acyclic graph (i.e. the directed acyclic graph before the change) has been discarded.

As to point (3), contrary to Applicant’s assertion the Picott prior art does teach “identifying the objects upon which a given object depends as those objects into which the given object passed itself as a requester during execution of a compute method of the given object”. The “dependency node A” represents the requester and a given object that is marked dirty when an identified object (dependency node B) changes. The “dependency node A” passes itself as requester when it is marked dirty as a result of a change to the (dependency node B) (“...evaluate me...” Col. 7 Ln. 31 – 47).

As to point (4), the error in not citing a reference for the claim limitation is an oversight and has been corrected. However, this correction does not change the ground of this rejection because the Picott prior art teaches this limitation as claim 1 also indicates.

As to point (5), as indicated above (point 2) the redraw request for the Picott prior art inherently includes the detachment of dependent objects from the dependency graph when a changes occurs because when a change or modification to an **existing DAG** occurs the redraw request is initialed in order to redraw or reconstruct the existing DAG. The redraw could only occur if the original directed acyclic graph is discarded or detached.

As to point (6), as claimed the limitation calls for the dynamic (automatic or runtime) calculation of dependency among objects when changes occur. The Razdow prior art covers this limitation because when a change (update/modification) occurs in a node, the dependency graph **automatically** updated.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Charles E. Anya whose telephone number is (571) 272-3757. The examiner can normally be reached on M-F (8:30-6:00) First Friday off.


If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, An Meng-Ai can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2194

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Charles E Anya
Examiner
Art Unit 2194

cea.



WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100